CLAIMS

What is claimed is:

1. A multi-processor computer system, comprising:

a plurality of processors coupled together to permit messages to be transmit from one processor to another processor;

at least one of said processor coupled to at least one input/output device; and

each processor having at least one timer that expires when a message is not sent from the processor in a predetermined amount of time.

2. The multi-processor computer system of claim 1 further wherein each processor can send a plurality of different message types to other of said processors and each includes a separate timer associated with each of said message types to expire when a message of the associated message type is not sent from the processor in a predetermined amount of time.

3. The multi-processor computer system of claim 2 wherein said message types include forward, request, response and input/output message types.

4. The multi-processor computer system of claim 1 wherein each processor includes a directory in-flight table that tracks pending messages and a timer associated with said directory in-flight table to monitor for traffic congestion.

5. The multi-processor computer system of claim 1 further including at least one register associated with each timer to permit the timer to be programmed.

6. The multi-processor computer system of claim 1 wherein each processor has at least one port connection to another processor and wherein each processor further includes a port timer associated with said inter-processor port connection.

7. The multi-processor computer system of claim 6 wherein each port timer increments if the associated port is being used to send messages.

8. The multi-processor computer system of claim 7 wherein each port timer is reset when a message is sent from the port.

9. The multi-processor computer system of claim 7 wherein each port timer is reset when it receives a signal from a processor that receives a message from the port that indicates that the receiving processor has freed up an entry in an input buffer.

10. A processor that can be coupled to other processors to form a multi-processor system and can exchange messages with other processors in the system, the processor comprising:

router logic that can be coupled to at least one other processor;

said router logic having at least one timer that expires when a message is not sent from the processor in a predetermined amount of time.

11. The processor of claim 10 further wherein each processor can send a plurality of different message types to other of said processors and each includes a separate timer associated with each

of said message types to expire when a message of the associated message type is not sent from the processor in a predetermined amount of time.

12. The processor of claim 11 wherein said message types include forward, request, response and input/output message types.

13. The processor of claim 10 wherein each processor includes a directory in-flight table that tracks pending messages and a timer associated with said directory in-flight table to monitor for traffic congestion.

14. The processor of claim 10 further including at least one register associated with each timer to permit the timer to be programmed.

15. The processor of claim 10 wherein each processor has at least one port connection to another processor and wherein each processor further includes a port timer associated with said inter-processor port connection.

16. The processor of claim 15 wherein each port timer increments if the associated port is being used to send messages.

17. The processor of claim 16 wherein each port timer is reset when a message is sent from the port.

18. The processor of claim 16 wherein each port timer is reset when it receives a signal from a processor that receives a message from the port that indicates that the receiving processor has freed up an entry in an input buffer.

19. A method of monitoring a computer system for traffic congestion, comprising:

starting a timer on the occurrence of a first predetermined event;

resetting the timer on the occurrence of a second predetermined event; and

if the timer is not reset and the timer expires, blocking further messages from being sent by a processor.

20. The method of claim 19 further including programming the timer.

21. The method of claim 19 wherein said first predetermined event is a buffer having at least one message in it.

22. The method of claim 19 wherein said second predetermined event is a message being sent from one processor to another.

23. A method of isolating failures in a multi-processor system, comprising:

(a) programming the system to have at least one partition selected from the group consisting of: hard partition, firm partition, semi-hard partition, and soft partition;

(b) detecting a failure in the system;

(c) blocking messages from being sent from one processor to another processor upon detecting said failure; and

(d) using timers to monitor the system for messages that are not able to be completed due to (c).